# Authors' Responses

- Paper ID: No. 25-TIE-1011
- Paper Title: Transparent Lifting Assistance Strategy for Arm Exoskeletons Using an Adaptive Torque Controller
- Authors: Xiuze Xia, Long Cheng, Lijun Han

- **First of all, We would like to express our gratitude to the AE, and the reviewers, for the time and effort that has been spent in processing our paper. The constructive comments from the editors and reviewers are very helpful in improving our paper. All comments have been addressed, and all modifications have been marked in red throughout the revised manuscript and this response letter.**

- **Responses to the Associate Editor**
  - ***Comment:*** *Five reviews are obtained, with several critical concerns. The reviewers feel that the novelty of the ARMPC and ANFIS-based adaptation is not well justified, and important design details and stability analysis lack clarity. Experimental validation is limited by a small sample size and insufficient comparison with advanced controllers. Practical concerns about torque reference generation and real-time feasibility were also raised. Additionally, inconsistencies in equations, unclear notations, and a lack of clarity in explanations reduce the overall readability and technical rigor.*

    **Thank you for the valuable feedback on our manuscript.** All the comments are immensely helpful in improving the clarity and quality of the paper. We have carefully considered each suggestion and made the relevant revisions accordingly. Please find the revised manuscript attached along with a detailed response to each reviewer's comments. And the main content of the revision is summarized blew.

    (1) By adding references and discussions of recent advanced studies, the advanced nature of ARMPC in exoskeleton assistive applications has been demonstrated;

    (2) More details of the controller and experimental setup have been added to enhance the completeness and the readability of the paper;

    (3) The stability of the nominal MPC and the bounded error of the ARMPC controller have been more thoroughly reasoned;

    (4) A comparison with Tube-based MPC has been included. Tube-based MPC is a widely used advanced control method in recent exoskeleton research;

    (5) Additional validation and analysis have been conducted under large disturbances and complex reference trajectories to further evaluate controller performance;

(6) More subjects have been added in the experiments, and statistical analysis has been added to improve the statistical significance of the results;

(7) An explanation has been added on how to ensure the controllers run in real time;

(8) The Introduction section has been revised to more logically present the research background and motivation of this study;

(9) Inconsistencies in equations, unclear notations, and other errors in the manuscript have been corrected, and the clarity of descriptions has been significantly improved.

**We sincerely appreciate the Editors and Reviewers for your time and effort in reviewing this paper. Thank you very much! If you have further questions, please feel free to let us know.**

- **Responses to Reviewer 1**
  - ***General Comments:*** *The manuscript presents a promising approach to exoskeleton-assisted lifting with validated torque tracking and user comfort. However, the technical rigor of the control framework and experimental scope require enhancement.*

    **Thank you for your valuable comments and suggestions.** In the revised manuscript, all the comments have been carefully answered and addressed.

  - ***Comment 1:*** *The proposed ARMPC framework and transparent assistance strategy are incremental improvements over existing methods. While the ANFIS-based parameter adaptation is novel in this context, its superiority over conventional gain-scheduling techniques is not thoroughly justified.*

    Thank you for your insightful comment. The primary reasons for employing ANFIS instead of traditional gain-scheduling in our ARMPC framework are as follows:

    First, conventional gain-scheduling techniques typically relies on pre-defined rules or scheduling tables, which may be suboptimal when system dynamics exhibit strong nonlinearities and disturbance influences. ANFIS combines fuzzy inference with neural network learning, enabling it to effectively capture complex and nonlinear input-output relationships without requiring explicit modeling or manual rule design. In this paper, the training data for ANFIS is constructed based on expert knowledge, where the input-output relationships inherently exhibit nonlinear characteristics. As a result, ANFIS is particularly well-suited for adapting the controller parameters in this context.

    Second, ANFIS's learning capability enables it to extract patterns from training data and apply them in unseen scenarios, providing superior adaptability compared to manually designed gain-scheduling. This is particularly advantageous in dynamic and unstructured human-robot interaction settings.

    Another key advantage of the ANFIS is its extensibility. The adaptation mechanism in ANFIS can be further enhanced through integration with other learning-based methods such as reinforcement learning (RL) or meta-learning. This compatibility facilitates the development of more advanced adaptive controllers capable of handling increasingly complex environments and tasks in future work.

    Based on the above considerations, this study ultimately chooses to employ the ANFIS neural network to achieve adaptive adjustment of the PI controller's parameters.

  - ***Comment 2:*** *The interaction torque model (linear state-space with disturbances) is practical but lacks originality compared to nonlinear models in recent works.*

    Thank you for your careful review. To the best of our knowledge, existing research on arm exoskeleton assistance has not yet established a physical model that treats the human-robot interaction torque as the state variable. Currently, nonlinear models of human-robot interac-

tion force are primarily applied in surgical robots. For example, Dominici *et al.* proposed a second-order hysteresis model for interaction force [1] and applied model predictive control to achieve precise force control during cardiac surgery. Similarly, Wijayarathne *et al.* developed an interaction force model with numerous physical parameters for a surgical robot [2]. Further examples of similar work can be found in the review article [3].

Although the approaches used in surgical robots have inspired this research, the torque modeling methods from those studies cannot be directly applied in exoskeleton control scenarios. In other words, the nonlinear modeling techniques they used are quite different from what we propose. The following will explain these differences from two perspectives.

First, both surgical robots and exoskeletons involve highly complex physical characteristics at the human-robot interaction interface. In surgical robots, interaction occurs between a rigid tip and soft human tissue, with a small contact area, making nonlinear modeling of interaction force feasible. In contrast, exoskeletons have a larger interaction interface area and typically use flexible materials such as straps for interaction, which makes direct nonlinear physical modeling of the interaction force less feasible. In our research, to address this challenge, we previously installed an SEA at the human-robot interaction port [4]. With this design, the SEA is essentially connected in series with the interaction port, so its output torque closely approximates the human-robot interaction torque. Therefore, by modeling the SEA output torque, we can effectively model the interaction torque.

Second, the interacted object for surgical robots is static. In contrast, in exoskeleton human-robot interaction processes, users perform voluntary movements. As a result, the interaction torque is influenced by the user's voluntary movements, meaning the model inevitably contains information about the user's voluntary movement trajectories, which further complicates the modeling process. To address this issue, we introduced an online arm movement trajectory prediction algorithm [5]. With this prediction algorithm, the proposed interaction torque model can be derived over time and applied to the RMPC.

To help readers better understand the differences between the torque model proposed in this study and the nonlinear models from the other research, we rewrote the Introduction section, adding relevant comparisons and analyses, as shown below in red.

[1] M. Dominici and R. Cortesao, "Model predictive control architectures with force feedback for robotic-assisted beating heart surgery," in *Proceedings of IEEE International Conference on Robotics and Automation*, Hong Kong, China, 2014, pp. 2276-2282.

[2] L. Wijayarathne, Z.Y. Zhou, Y. Zhao, and F.H. Hammond, "Real-time deformable-contact-aware model predictive control for force-modulated manipulation," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3549-3566, 2023.

[3] A.S.S. Abadi, A. Ordys, K. Kukielka, and B. Pierscionek, "Review on challenges for robotic eye surgery; surgical systems, technologies, cost-effectiveness, and controllers," International

Journal of Medical Robotics and Computer Assisted Surgery, vol. 19, no. 4, pp. 1-18, 2023.
[4] X. Xia, L. Han, H. Li, Y. Zhang, Z. Liu, L. Cheng, "A compliant elbow exoskeleton with an SEA at interaction port," in *Proceedings of the 2023 International Conference on Neural Information Processing*, Changsha, China, vol. 14450, pp. 146-157, 2023.
[5] X. Xia, L. Cheng, M. Ma, H. Zhang, L. Han, and H. Li, "Neurologically inspired transparent interaction for wearable exoskeletons," *IEEE/ASME Transactions on Mechatronics*, DOI: 10.1109/TMECH.2024.3492010, early access.

In addition to exoskeletons, surgical robot control scenarios also treat the interaction force as the controlled variable. In such scenarios, precise control of the human-robot interaction force is crucial. Therefore, the physical characteristics of the human-robot interaction port are typically modeled. It is worth mentioning that the concept of the human-robot interaction port typically refers to the area where the human and the robot make contact [24]. For example, Dominici *et al.* proposed a second-order hysteresis model for the interaction force [25] and applied model predictive control to achieve precise force control during cardiac surgery. Similarly, Wijayarathne *et al.* developed an interaction force model with numerous physical parameters for a surgical robot [26]. The main drawbacks of these models are their high complexity and the need to identify numerous physical parameters, which can reduce the real-time control frequency of the algorithm.

However, even if the control frequency issue can be resolved, applying the interaction force modeling approaches used in surgical robots to exoskeletons still faces two key challenges. First, surgical robots interact with flexible human tissue through a rigid tip, resulting in a small contact area that makes nonlinear modeling of the human-robot interaction force feasible. In contrast, the human-exoskeleton interaction port covers a larger area and typically involves flexible materials such as straps, making direct nonlinear physical modeling of the interaction force much less feasible. Second, the interacted object in surgical robots is static. In exoskeleton applications, however, the user actively moves. In this case, accurately modeling the human-robot interaction force would require predicting the user's movements.

– ***Comment 3:*** *The ANFIS design (Section IV.B) requires more clarity: How were the membership functions initialized? What training data were used, and how was overfitting avoided?*

Thank you for your careful review. The membership function $\mu_{ij}$ was initialized using the Fuzzy C-Means (FCM) algorithm. The expression for $\mu_{ij}$ is given by:

$$\mu_{ij} = \exp\left(-\left(\frac{e_i - c_{ij}}{\sigma_{ij}}\right)^2\right),$$

where the subscript $i$ denotes the $i$-th input variable of ANFIS ($i = 1, 2$), and $j$ represents the $j$-th fuzzy rule ($j = 1, 2, 3$). $c_{ij}$ denotes the mean of the $j$-th membership function of the $i$-th input, and $\sigma_{ij}$ represents the standard deviation. The parameters to be identified are $cij$ and $\sigma_{ij}$.

In this paper, each input variable has three fuzzy rules. Therefore, the number of clusters in FCM was set to 3. The FCM algorithm provides three cluster centers and the corresponding membership matrix. The initial values of $c_{ij}$ are directly set to the calculated cluster centers. The initial values of $\sigma_{ij}$ are calculated using the following formula:

$$\sigma_{ij}^2 = \frac{\sum_{t=1}^{T} \lambda_{ijt}^2 \left(\chi_{it} - c_{ij}\right)^2}{\sum_{t=1}^{T} \lambda_{ijt}^2},$$

where $\lambda_{ijt}$ is the membership degree of the $t$-th training data point to the $j$-th fuzzy cluster, which can be obtained from the membership matrix. $\chi_{it}$ represents the $i$-th component of the $t$-th training data point.

The above describes the initialization method of the membership function $\mu_{ij}$. To enhance the completeness of the paper, we have supplemented the original manuscript, as shown below.

$c_{ij}$ denotes the mean of the $j$-th membership function of the $i$-th input, and $\sigma_{ij}$ represents the standard deviation. The parameters $c_{ij}$ and $\sigma_{ij}$ were initialized using the fuzzy C-means (FCM) algorithm. Considering that the proposed ANFIS includes 3 fuzzy rules, the number of clusters in FCM was set to 3. After executing the FCM algorithm, three cluster centers and the corresponding membership matrix can be obtained. The mean values $c_{ij}$ are directly set to the calculated cluster centers. The values of $\sigma_{ij}$ are calculated using the following formula:

$$\sigma_{ij}^2 = \frac{\sum_{t=1}^{N} \lambda_{ijt}^2 \left(\chi_{it} - c_{ij}\right)^2}{\sum_{t=1}^{N} \lambda_{ijt}^2}, \tag{44}$$

where $\lambda_{ijt}$ is the membership degree of the $t$-th training data point to the $j$-th fuzzy cluster, which can be obtained from the membership matrix. $\chi_{it}$ represents the $i$-th component of the $t$-th training data point.

To generate the training data, the values of $K_P$ and $K_I$ are recorded for different error conditions based on the expert knowledge. Then the sparse training data are augmented with smooth transition functions. The aforementioned different error conditions include what we consider the most extreme cases—*i.e.*, the cases with the maximum and minimum errors. Furthermore, sparse sampling of these different error conditions is achieved through non-uniform sampling, where more sampling points are selected when the error is smaller. This approach allows us to design the values of $K_P$ and $K_I$ based on expert knowledge.

It is worth mentioning that the $K_P$ and $K_I$ used in this study are required to strictly satisfy condition (48). Condition (48) essentially provides the necessary and sufficient condition for the stability of the closed-loop discrete-time linear system with a PI controller. To ensure that all points in the training set satisfy (48), the $K_P$ and $K_I$ designed based on expert knowledge are conservative. In addition, we extracted the extreme points from the augmented training

set and confirmed that these points also satisfy (48), thereby ensuring that the entire training set meets this condition.

To improve the completeness of the paper, the revised manuscript includes supplementary explanations regarding the selection of error conditions and the constraints considered when designing the training data, as shown below.

To generate the training data, the values of $K_P$ and $K_I$ are recorded for different error conditions based on the expert knowledge. The aforementioned different error conditions include what we consider the most extreme cases—*i.e.*, the cases with the maximum and minimum errors. Furthermore, sparse sampling is achieved through non-uniform sampling, where more sampling points are selected when the error is smaller. Based on these sampling points and expert knowledge, sparse training data points are obtained. The sparse training data points are then augmented with smooth transition functions.

To guarantee the error boundedness of the controller, parameters $K_P$ and $K_I$ in the training set are required to strictly satisfy the following conditions:

$$|\mathrm{eig}\,(A_c)| < 1, \tag{48}$$

$$A_c = \begin{bmatrix} A - \bar{B}K_P & -\bar{B}K_I \\ A - \bar{B}K_P & -\bar{B}K_I + I \end{bmatrix}, \tag{49}$$

where $I$ is the identity matrix. Condition (48) essentially provides the necessary and sufficient condition for the stability of the closed-loop discrete-time linear system with a proportional-integral (PI) controller. To ensure that all points in the training set satisfy (48), the $K_P$ and $K_I$ designed based on expert knowledge are conservative. In addition, we extracted the extreme points from the augmented training set and confirmed that these points also satisfy (48), thereby ensuring that the entire training set meets this condition.

To prevent overfitting, we limited the ANFIS model to only three fuzzy rules, thereby controlling the model complexity and reducing the risk of overfitting. Additionally, we monitored the validation error during training and applied early stopping when the validation error began to increase while the training error continued to decrease, thus further mitigating the risk of overfitting.

– ***Comment 4:*** *The stability analysis of ARMPC needs to be strengthened (e.g. via Lyapunov theory).*

Thank you for your careful review. In the previous version, the proofs for the stability of the nominal part of the ARMPC and the overall error boundedness were not detailed enough. In the revised manuscript, we have added relative supplementary analysis to enhance the rigor of the paper and make it easier for readers to understand. The revised stability proof section combining Lyapunov theory is shown below.

According to the optimality of $V^*(k)$, the following inequality can be obtained.

$$V^*(k+1) - V^*(k) \leq J(k+1) - V^*(k). \tag{31}$$

Define the optimal control input sequence in $V^*(k)$ as $\left[v_1^*(0|k), \cdots, v_1^*(N-1|k)\right]$, and the control input sequence in $J(k+1)$ as $\left[v_1^*(1|k), \cdots, v_1^*(N-1|k), v_1(N|k)\right]$, where $v_1(N|k) = K\mathbf{z}(N|k)$. Then (31) can be further simplified

$$\begin{aligned} &V^*(k+1) - V^*(k) \leq J(k+1) - V^*(k) \\ &= \|v_1(N|k)\|_R^2 + \|\mathbf{z}(N|k)\|_Q^2 + \|\mathbf{z}(N+1|k)\|_P^2 \\ &- \|\mathbf{z}(N|k)\|_P^2 - \|\mathbf{z}(1|k)\|_Q^2 - \left\|v_1^*(0|k)\right\|_R^2. \end{aligned} \tag{32}$$

Multiply $\mathbf{z}^T(N|k)$ on the left side and $\mathbf{z}(N|k)$ on the right side of the the discrete-time Lyapunov equation (25), we can get

$$\begin{aligned} &\mathbf{z}^T(N|k) P\mathbf{z}(N|k) - \left[\left(A - \bar{B}K\right)\mathbf{z}(N|k)\right]^T P\left(A - \bar{B}K\right)\mathbf{z}(N|k) \\ &= \mathbf{z}^T(N|k) Q\mathbf{z}(N|k) + \left[K\mathbf{z}(N|k)\right]^T RK\mathbf{z}(N|k)\,\mathbf{z}(N|k). \end{aligned} \tag{33}$$

Considering that $v_1(N|k) = K\mathbf{z}(N|k)$, we can get

$$\left(A - \bar{B}K\right)\mathbf{z}(N|k) = \mathbf{z}(N+1|k), \tag{34}$$

then (33) can be simplified as

$$\|\mathbf{z}(N|k)\|_P^2 - \|\mathbf{z}(N+1|k)\|_P^2 = \|\mathbf{z}(N|k)\|_Q^2 + \|v_1(N|k)\|_R^2. \tag{35}$$

Substituting (35) into (32) yields

$$V^*(k+1) - V^*(k) \leq -\|\mathbf{z}(1|k)\|_Q^2 - \|v_1(0|k)\|_R^2 \leq 0. \tag{36}$$

Therefore, the MPC designed for the nominal model is stable.

– ***Comment 5:*** *The comparison with impedance control (IC) is justified, but the absence of comparisons with other advanced torque controllers weakens the argument for ARMPC's superiority.*

Thank you for your careful review. In recent years, advanced exoskeleton robot control research has extensively employed model predictive control [6], [7], [8], [9]. Most of these studies also utilize robust model predictive control algorithms. For example, Xu *et al.*, in their 2025 paper published in *IEEE Transactions on Cybernetics*, developed an RMPC method based on a linearized system dynamics model [6]. This demonstrates that RMPC is a commonly used control algorithm in advanced exoskeleton research. In addition, some studies have explored more complex Tube-based MPC frameworks, such as the exoskeleton control approach introduced by Bao *et al.* in their 2021 work published in *IEEE Transactions on Control Systems Technology* [9].

Compared to these studies, our work has two main distinctions. First, we propose a dynamic model of the human-robot interaction torque and integrate it within the MPC framework. Second, we develop an adaptive robust model predictive control algorithm to enhance the controller's accuracy. To further benchmark our approach against state-of-the-art research, we have added the simulation and the experiment using Tube-based MPC for torque control in the revised manuscript, along with a detailed discussion on the advances of our proposed algorithm.

[6] L.R. Xu, Z.J. Li, G.X. Li, and L.J. Jin, "Robust model predictive control of a gait rehabilitation exoskeleton with whole body motion planning and neuro-dynamics optimization," *IEEE Transactions on Cybernetics*, DOI: 10.1109/TCYB.2025.3545064, early access.
[7] X. Jin and J. Guo, "Disturbance rejection model predictive control of lower limb rehabilitation exoskeleton," *Scientific Reports*, vol. 13, no. 19463, 2023.
[8] Z.Y. Sun, A. Iyer, K. Lambeth, C. Cleveland, and N. Sharma, "Knee extension tracking and fatigue regulation results using a robust MPC approach in a hybrid exoskeleton," *Control Engineering Practice*, vol. 141, no. 105717, 2023.
[9] X.F. Bao, Z.Y. Sheng, B.E. Dicianno, and N. Sharma, "A tube-based model predictive control method to regulate a knee joint with functional electrical stimulation and electric motor assist," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 5, 2021.

The analysis of the algorithm's advanced nature is presented as follows:

In recent exoskeleton research, robust model predictive controllers (RMPC) have been widely used [35]-[37]. For example, Xu *et al.* designed an RMPC based on a linearized system dynamic model [35]. Bao *et al.* developed a control system for a knee exoskeleton using a Tube-based MPC approach [37]. Compared to these studies, our work differs in two main ways. First, we propose a dynamic model for human-robot interaction torque and incorporate it into the model predictive control framework. Second, we integrate an adaptive strategy into the RMPC to enhance the controller's accuracy. Given the advanced nature of these algorithms, we included a comparative simulation with both RMPC and Tube-based MPC to highlight the advantages of our proposed ARMPC. The RMPC used for comparison is similar to the proposed ARMPC, except that its $\boldsymbol{K_P}$ and $\boldsymbol{K_I}$ in (40) are set to appropriate fixed values. The Tube-based MPC used for comparison is a more robust MPC algorithm that leverages error tubes and tightened constraints, with specific design methods detailed in [37].

In addition, impedance control (IC) is one of the most widely used force control algorithm in exoskeleton robots. It can achieve compliant human-robot interaction without the need to model the interaction forces, making it widely adopted in exoskeleton assistance [38], [39]. Consequently, this study also uses impedance control as a comparison algorithm.

The supplementary simulation is as follows:

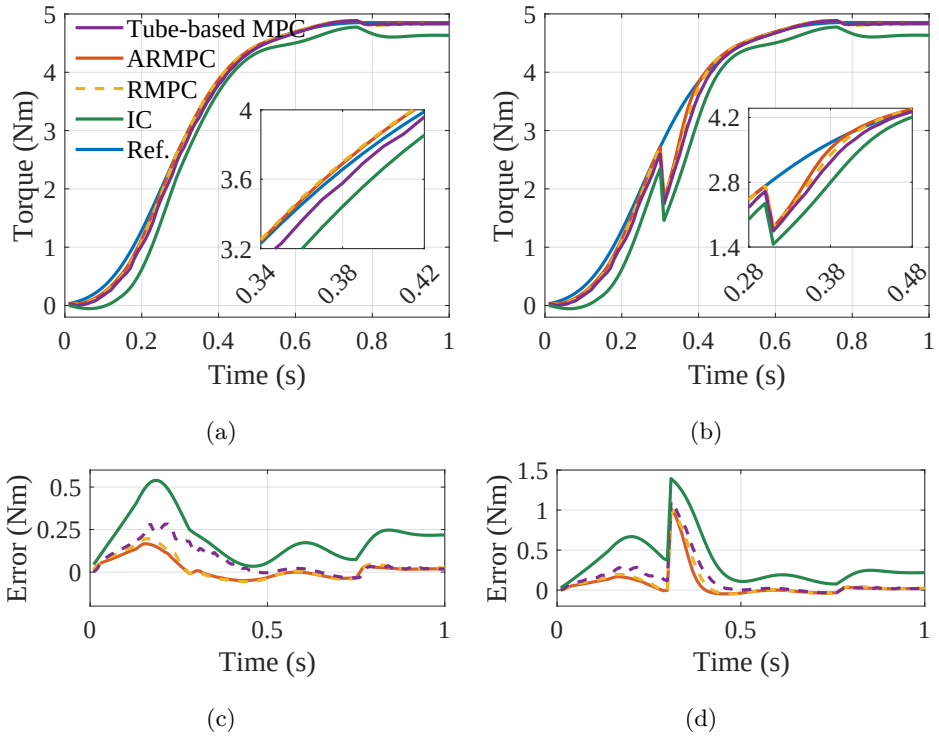



Fig. 6: The performance of the proposed torque controllers in simulation. (a) The torque tracking performance of the controllers. (b) The torque tracking performance in the presence of the sudden disturbance. (c) The torque tracking error of the controllers. (d) The torque tracking error in the presence of the sudden disturbance.

The simulation results are shown in Fig. 6(a) and Fig. 6(c). The mean absolute errors in torque trajectory tracking for ARMPC, RMPC, Tube-based MPC, and IC are 0.043 Nm, 0.052 Nm, 0.661 Nm, and 0.203 Nm, respectively. It can be observed that the control algorithms based on the interaction torque model (ARMPC, RMPC, and Tube-based MPC) exhibit higher control accuracy, demonstrating that the proposed model effectively captures the characteristics of the human-robot interaction torque. Moreover, ARMPC achieves the highest control accuracy, indicating that the adaptive algorithm provides more reasonable feedback controller parameters under different error states than RMPC. This improvement is beneficial for the design and validation of exoskeleton assistance strategies. It is also worth noting that the Tube-based MPC is a more conservative algorithm that sacrifices some accuracy to improve controller robustness.

The supplementary experiment is as follows:

From Fig. 9, it can be seen that the experimental results align closely with the simulation in Section V, with ARMPC showing the best performance. This further confirms that the

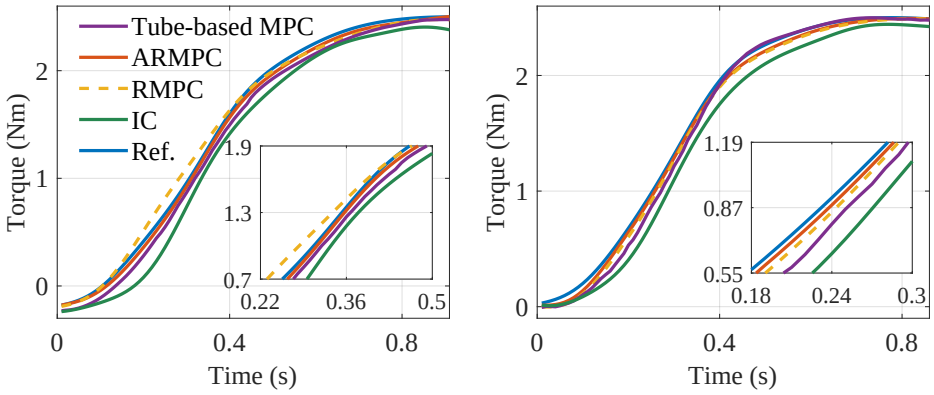



Fig. 9: A preliminary performance comparison among the three controllers during the lifting assistance tasks. (a) Subject 1. (b) Subject 2.

introduction of the interaction torque model and the adaptive algorithm can effectively enhance the precision of the interaction torque controller. Video 1 in the supplementary materials demonstrates the same results. It is worth noting that there is often a trade-off between the controller precision and the robustness. Theoretically, Tube-based MPC improves robustness at the expense of control accuracy. Similarly, the adaptive algorithm proposed in this paper may also reduce the controller's robustness to some extent. Since it is difficult to precisely control the magnitude of external disturbance torques in physical experiments, the controller's response to sudden, large disturbances is discussed through simulations in Section V. In the real-world experiments, the exoskeleton assistance tasks were limited to load-lifting activities, and all participants were healthy individuals. As a result, occurrences of sudden and significant disturbances were extremely rare. Moreover, to ensure user safety, the human-robot interaction force was monitored in real time during the experiments. If the interaction force exceeded a predefined threshold, the exoskeleton would be shut down to prevent potential harm.

– ***Comment 6:*** *The sample size (6 participants) is small, limiting statistical significance.*

Thank you for your careful review. To improve the statistical significance of our research, we have added four new participants (including one female and three male). In the revised manuscript, all experiments involve ten participants. The added data and corresponding analysis have been fully supplemented in the revised manuscript. In addition, more statistical analyses have been added to the revised manuscript, as shown below.

Considering that the results in Fig. 9 are approximate, for each controller, the control errors from ten subjects were combined and visualized using box plots, as shown in Fig.

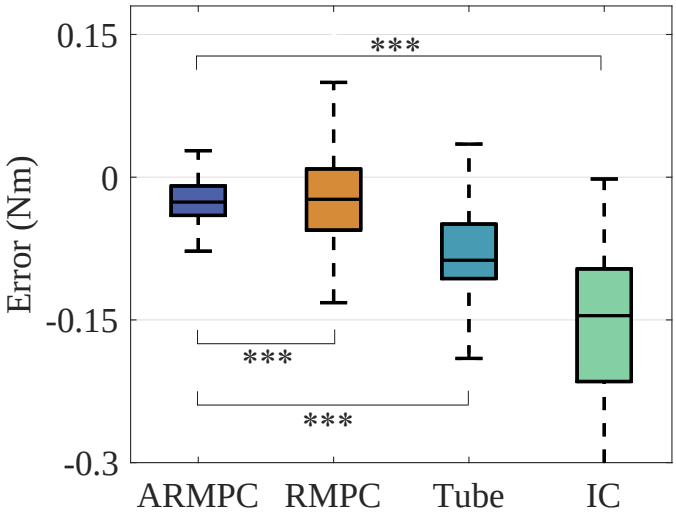



Fig. 10: Torque tracking errors comparison. The 'Tube' refers to the Tube-based MPC.

10. *** in Fig. 10 indicates a statistically significant difference between the two groups (t-test, $p < 0.001$). It can be observed that the interaction torque control error of ARMPC is significantly lower than that of the other controllers.

- **Responses to Reviewer 2**
  - ***General Comments:*** *In this paper, a human-computer interaction torque modeling method is proposed, and a torque controller is designed based on an adaptive robust model predictive control algorithm for reciprocal torque trajectory tracking. Although the feasibility of the method is verified, there are still some suggestions to be adopted:*

    **Thank you for your valuable comments and suggestions.** In the revised manuscript, all the comments have been carefully answered and addressed.

  - ***Comment 1:*** *The design of the nominal MPC controller involves terminal constraint sets; please carefully explain the terminal set calculation method.*

    Thank you for your valuable comment. In our paper, the nominal MPC controller incorporates terminal constraint sets to ensure stability and feasibility. Here, we provide a detailed explanation of the method used to compute the terminal set:

    The terminal constraint set, denoted as $\mathcal{Z}_f$, is chosen to be a robust positively invariant (RPI) set under a local control law $K\mathbf{z}$, where $K$ is a stabilizing feedback gain satisfying (26) in the revised manuscript. This set ensures that if the system state reaches $\mathcal{Z}_f$ at the end of the prediction horizon, the closed-loop system will remain within constraints thereafter.

    In theory, $\mathcal{Z}_f$ is computed as the maximal positively invariant (MPI) set for the system dynamics under $u' = K\mathbf{z}$ and in the presence of state and input constraints. This involves iteratively applying the set dynamics until convergence:

    $$\mathcal{Z}_f^{(i+1)} = \left\{\mathbf{z} \middle| A\mathbf{z} + \bar{B}K\mathbf{z} \in \mathcal{Z}_f^{(i)}, \mathbf{z}(n|k) \in \mathcal{Z}, K\mathbf{z} \in \mathcal{V}_1\right\},$$

    where $\mathcal{Z}$ is the set of nominal state constraints, and $\mathcal{V}_1$ is the set of control input constraints. The superscript $(i+1)$ represents the iteration number.

    In practice, we approximate $\mathcal{Z}_f$ using an ellipsoidal outer approximation of the invariant set, which is computed via the MPT3 Toolbox in MATLAB.

  - ***Comment 2:*** *The article suggests that controller stability can be satisfied by satisfying (46) and (47), please prove it.*

    Thank you for your careful review and valuable comment. First, there was a misstatement in the previous manuscript, where (46) and (47) (now (48) and (49) in the revised manuscript) were claimed to guarantee the stability of the ARMPC, whereas they actually ensure the boundedness of the torque tracking error. Second, the original proof of the controller's error boundedness based on equations (46) and (47) was rather brief. In the revised manuscript, we have expanded this section to facilitate readers' understanding, as shown below.

    With (48) and (49), the torque tracking error $\mathbf{e}(k)$ is bounded, and the proof is shown below:

Substitute (40) into (39) yields

$$\Xi(k+1) = A\Xi(k) - \bar{B}K_P(k)\Xi(k) - \bar{B}K_I(k)\Upsilon(k) + \zeta(k) - w(k). \tag{50}$$

Considering that

$$\Upsilon(k+1) = \Upsilon(k) + \Xi(k+1), \tag{51}$$

combine (50) and (51), we can obtain an expanded state space equation.

$$\begin{bmatrix}\Xi(k+1)\\ \Upsilon(k+1)\end{bmatrix} = \begin{bmatrix}A - \bar{B}K_P & -\bar{B}K_I\\ A - \bar{B}K_P & -\bar{B}K_I + I\end{bmatrix}\begin{bmatrix}\Xi(k)\\ \Upsilon(k)\end{bmatrix} + \begin{bmatrix}I\\ I\end{bmatrix}\zeta(k) - \begin{bmatrix}I\\ I\end{bmatrix}w(k). \tag{52}$$

The scenario of our research is the arm lifting movement assistance. In this context, the arm performs the point-to-point movement, meaning that the arm's trajectory ultimately converges to a fixed value. Therefore, $\zeta(k)$ is a bounded variable, and there exists a $t_0 > 0$ such that for $k > t_0$, $\zeta(k)$ becomes constant. Since $w \in \mathcal{W}$ is bounded, the difference $\zeta(k) - w(k)$ is also bounded. Let

$$\varphi(k) = \begin{bmatrix}I\\ I\end{bmatrix}\zeta(k) - \begin{bmatrix}I\\ I\end{bmatrix}w(k) \in \Phi, \tag{53}$$

where $\Phi$ is a bounded set. Then (51) can be rewritten as

$$\Gamma(k+1) = A_c\Gamma(k) + \varphi(k), \tag{54}$$

$$\Gamma(k) = \begin{bmatrix}\Xi(k)\\ \Upsilon(k)\end{bmatrix}. \tag{55}$$

Since $z(0) = e(0)$, we have $\Xi(0) = e(0) - z(0) = 0$ and $\Gamma(0) = 0$. Then,

$$\Gamma(k) = A_c^k\Gamma(0) + \sum_{i=1}^{k} {A_c}^{i-1}\varphi(k-i) = \sum_{i=1}^{k} {A_c}^{i-1}\varphi(k-i)$$

$$\in \sum_{i=1}^{k} {A_c}^{i-1}\Phi \subset \sum_{i=1}^{\infty} {A_c}^{i-1}\Phi \triangleq \Psi. \tag{56}$$

Considering that $|\mathrm{eig}(A_c)| < 1$, $\Psi = \sum_{i=1}^{\infty} {A_c}^{i-1}\Phi$ is finite, so $\Gamma(k)$ is bounded, and $\Xi(k)$ is also bounded. The stability of the MPC for the nominal model is proved in Section IV.A, which ensures that $\mathbf{z}(k)$ converges, and that $\mathbf{e}(k) = \mathbf{z}(k) + \Xi(k)$ remains bounded.

– ***Comment 3:*** *The authors should explain the expression of fuzzy rules in neural networks and the method of determining the corresponding parameters to enhance the readability of the article.*

Thank you for your careful review. The expressions of the fuzzy rules in the ANFIS neural network have been added to the revised manuscript. In addition, the initialization and training processes of the relevant parameters have been introduced, as shown below.

The expression of the $j$-th fuzzy rule is shown below.

$$R_j\text{: If } e_1 \text{ is } A_{1j} \text{ and } e_2 \text{ is } A_{2j},$$
$$\text{then } y_{1j} = p_{1j}e_1 + q_{1j}e_2 + r_{1j} \text{ and } y_{2j} = p_{2j}e_1 + q_{2j}e_2 + r_{2j}, \tag{45}$$

where $A_{1j}$ and $A_{2j}$ denote the fuzzy sets of the $j$-th fuzzy rule. $y_{1j}$ and $y_{2j}$ denote the outputs. $p_{ij}$, $q_{ij}$, and $r_{ij}$ are parameters, and their initial values are randomly initialized. During training, the antecedent parameters ($c_{ij}$ and $\sigma_{ij}$) are updated using backpropagation and gradient descent, while the consequent parameters ($p_{ij}$, $q_{ij}$, and $r_{ij}$) are optimized using the least squares method.

– ***Comment 4:*** *Regarding the training process of ANFIS network, the authors are requested to explain how the error conditions are selected to train the dataset.*

Thank you for your careful review. In this paper, the values of $K_P$ and $K_I$ are recorded for different error conditions based on the expert knowledge. Then the sparse training data are augmented with smooth transition functions. The aforementioned different error conditions include what we consider the most extreme cases—*i.e.*, the cases with the maximum and minimum errors. Furthermore, sparse sampling of these different error conditions is achieved through non-uniform sampling, where more sampling points are selected when the error is smaller. This approach allows us to design the values of$K_P$ and $K_I$ based on expert knowledge.

It is worth mentioning that the $K_P$ and $K_I$ used in this study are required to strictly satisfy condition (48). Condition (48) essentially provides the necessary and sufficient condition for the stability of the closed-loop discrete-time linear system with a PI controller. To ensure that all points in the training set satisfy (48), the $K_P$ and $K_I$ designed based on expert knowledge are conservative. In addition, we extracted the extreme points from the augmented training set and confirmed that these points also satisfy (48), thereby ensuring that the entire training set meets this condition.

To improve the completeness of the paper, the revised manuscript includes supplementary explanations regarding the selection of error conditions and the constraints considered when designing the training data, as shown below.

To generate the training data, the values of $K_P$ and $K_I$ are recorded for different error conditions based on the expert knowledge. The aforementioned different error conditions include what we consider the most extreme cases—*i.e.*, the cases with the maximum and minimum errors. Furthermore, sparse sampling is achieved through non-uniform sampling, where more sampling points are selected when the error is smaller. The sparse training data are then augmented with smooth transition functions.

To guarantee the error boundedness of the controller, parameters $K_P$ and $K_I$ in the training set are required to strictly satisfy the following conditions:

$$|\text{eig}\,(A_c)| < 1, \tag{48}$$

$$A_c = \begin{bmatrix} A - \bar{B}K_P & -\bar{B}K_I \\ A - \bar{B}K_P & -\bar{B}K_I + I \end{bmatrix}, \tag{49}$$

where $I$ is the identity matrix. Condition (48) essentially provides the necessary and sufficient condition for the stability of the closed-loop discrete-time linear system with a proportional-integral (PI) controller. To ensure that all points in the training set satisfy (48), the $K_P$ and $K_I$ designed based on expert knowledge are conservative. In addition, we extracted the extreme points from the augmented training set and confirmed that these points also satisfy (48), thereby ensuring that the entire training set meets this condition.

– ***Comment 5:*** *In the section on simulation and experiment, only the comparison with RMPC and IC is made, and no comparison is made with the advanced methods in recent years.*

Thank you for your careful review. In recent years, advanced exoskeleton robot control research has extensively employed model predictive control [1], [2], [3], [4]. Most of these studies also utilize robust model predictive control algorithms. For example, Xu *et al.*, in their 2025 paper published in *IEEE Transactions on Cybernetics*, developed an RMPC method based on a linearized system dynamics model [1]. This demonstrates that RMPC is a commonly used control algorithm in advanced exoskeleton research. In addition, some studies have explored more complex Tube-based MPC frameworks, such as the exoskeleton control approach introduced by Bao *et al.* in their 2021 work published in *IEEE Transactions on Control Systems Technology* [4].

Compared to these studies, our work has two main distinctions. First, we propose a dynamic model of the human-robot interaction torque and integrate it within the MPC framework. Second, we develop an adaptive robust model predictive control algorithm to enhance the controller's accuracy. To further benchmark our approach against state-of-the-art research, we have added the simulation and the experiment using Tube-based MPC for torque control in the revised manuscript, along with a detailed discussion on the advances of our proposed algorithm.

[1] L.R. Xu, Z.J. Li, G.X. Li, and L.J. Jin, "Robust model predictive control of a gait rehabilitation exoskeleton with whole body motion planning and neuro-dynamics optimization," *IEEE Transactions on Cybernetics*, DOI: 10.1109/TCYB.2025.3545064, early access.
[2] X. Jin and J. Guo, "Disturbance rejection model predictive control of lower limb rehabilitation exoskeleton," *Scientific Reports*, vol. 13, no. 19463, 2023.
[3] Z.Y. Sun, A. Iyer, K. Lambeth, C. Cleveland, and N. Sharma, "Knee extension tracking and fatigue regulation results using a robust MPC approach in a hybrid exoskeleton," *Control Engineering Practice*, vol. 141, no. 105717, 2023.
[4] X.F. Bao, Z.Y. Sheng, B.E. Dicianno, and N. Sharma, "A tube-based model predictive control method to regulate a knee joint with functional electrical stimulation and electric

motor assist," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 5, 2021.

The analysis of the algorithm's advanced nature is presented as follows:

In recent exoskeleton research, robust model predictive controllers (RMPC) have been widely used [35]-[37]. For example, Xu *et al.* designed an RMPC based on a linearized system dynamic model [35]. Bao *et al.* developed a control system for a knee exoskeleton using a Tube-based MPC approach [37]. Compared to these studies, our work differs in two main ways. First, we propose a dynamic model for human-robot interaction torque and incorporate it into the model predictive control framework. Second, we integrate an adaptive strategy into the RMPC to enhance the controller's accuracy. Given the advanced nature of these algorithms, we included a comparative simulation with both RMPC and Tube-based MPC to highlight the advantages of our proposed ARMPC. The RMPC used for comparison is similar to the proposed ARMPC, except that its $\boldsymbol{K_P}$ and $\boldsymbol{K_I}$ in (40) are set to appropriate fixed values. The Tube-based MPC used for comparison is a more robust MPC algorithm that leverages error tubes and tightened constraints, with specific design methods detailed in [37].

In addition, impedance control (IC) is one of the most widely used force control algorithm in exoskeleton robots. It can achieve compliant human-robot interaction without the need to model the interaction forces, making it widely adopted in exoskeleton assistance [38], [39]. Consequently, this study also uses impedance control as a comparison algorithm.

The supplementary simulation is as follows:

The simulation results are shown in Fig. 6(a) and Fig. 6(c). The mean absolute errors in torque trajectory tracking for ARMPC, RMPC, Tube-based MPC, and IC are 0.043 Nm, 0.052 Nm, 0.661 Nm, and 0.203 Nm, respectively. It can be observed that the control algorithms based on the interaction torque model (ARMPC, RMPC, and Tube-based MPC) exhibit higher control accuracy, demonstrating that the proposed model effectively captures the characteristics of the human-robot interaction torque. Moreover, ARMPC achieves the highest control accuracy, indicating that the adaptive algorithm provides more reasonable feedback controller parameters under different error states than RMPC. This improvement is beneficial for the design and validation of exoskeleton assistance strategies. It is also worth noting that the Tube-based MPC is a more conservative algorithm that sacrifices some accuracy to improve controller robustness.

The supplementary experiment is as follows:

From Fig. 9, it can be seen that the experimental results align closely with the simulation in Section V, with ARMPC showing the best performance. This further confirms that the introduction of the interaction torque model and the adaptive algorithm can effectively enhance the precision of the interaction torque controller. Video 1 in the supplementary

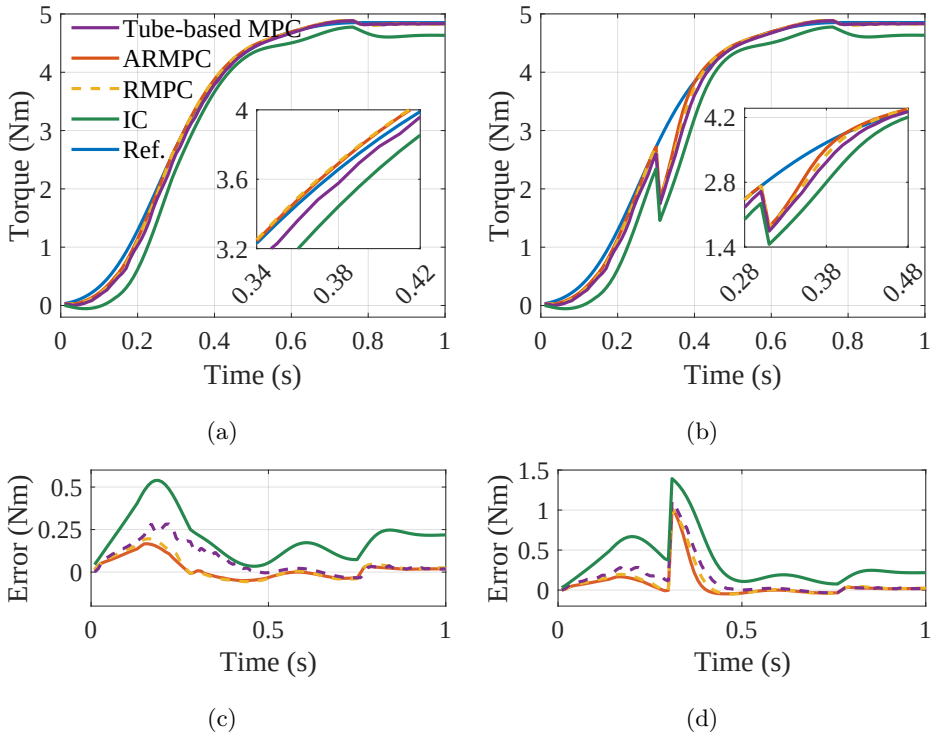



Fig. 6: The performance of the proposed torque controllers in simulation. (a) The torque tracking performance of the controllers. (b) The torque tracking performance in the presence of the sudden disturbance. (c) The torque tracking error of the controllers. (d) The torque tracking error in the presence of the sudden disturbance.

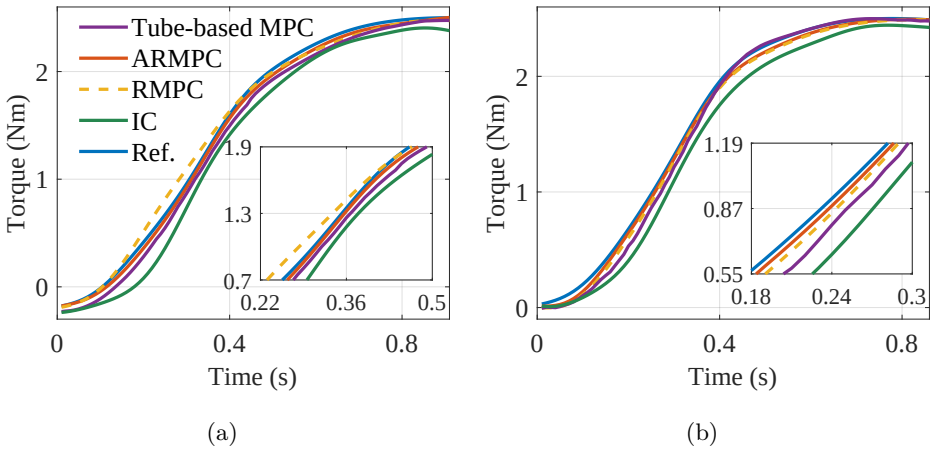



Fig. 9: A preliminary performance comparison among the three controllers during the lifting assistance tasks. (a) Subject 1. (b) Subject 2.

materials demonstrates the same results. It is worth noting that there is often a trade-off between the controller precision and the robustness. Theoretically, Tube-based MPC improves robustness at the expense of control accuracy. Similarly, the adaptive algorithm proposed in this paper may also reduce the controller's robustness to some extent. Since it is difficult to precisely control the magnitude of external disturbance torques in physical experiments, the controller's response to sudden, large disturbances is discussed through simulations in Section V. In the real-world experiments, the exoskeleton assistance tasks were limited to load-lifting activities, and all participants were healthy individuals. As a result, occurrences of sudden and significant disturbances were extremely rare. Moreover, to ensure user safety, the human-robot interaction force was monitored in real time during the experiments. If the interaction force exceeded a predefined threshold, the exoskeleton would be shut down to prevent potential harm.

– ***Comment 6:*** *The practical application of this research should focus on real-time; please discuss the real-time and computational efficiency of the method proposed in this paper.*

Thank you for your careful review. In the practical experiments, considering that algorithms such as RMPC involve constrained optimization problems, it is essential to ensure their real-time performance and computational efficiency. In this study, the control platform used was the dSPACE 1202 (MicroLabBox), and all controllers were executed online at a control frequency of 100 Hz. In practical implementation, we tuned the controllers' parameters to ensure real-time operation at 100 Hz. The main parameters tuned were the prediction horizons of the ARMPC, RMPC, and Tube-based MPC. Through tuning, we found that when the prediction horizon was set to more than 10, the program failed to execute properly. However, when it was set to 8, the program could run stably in real time. To improve the completeness of the manuscript, we have added a description of the control platform, as well as an explanation of how real-time performance was ensured. The added explanation is as follows:

The control platform used in the experiments was the dSPACE 1202 (MicroLabBox). All controllers were executed online at a frequency of 100 Hz. Considering that ARMPC, RMPC, and Tube-based MPC involve constrained optimization problems, to ensure real-time execution, their prediction horizons were carefully designed and set to 8 in practical applications.

- **Responses to Reviewer 3**
  - ***General Comments:*** *For the torque reference trajectory tracking problem, this paper proposes a ANFIS based ARMPC control scheme. The theoretical analysis is rigorous. And the experimental results are convincing because of the videos. However, there are some concerns that need to be addressed in this manuscript.*

    **Thank you for your valuable comments and suggestions.** In the revised manuscript, all the comments have been carefully answered and addressed.

  - ***Comment 1:*** *The first contribution is the proposal of the human-robot interaction torque modeling method. Has no one done similar research before? What's the difference between your study and theirs?*

    Thank you for your careful review. To the best of our knowledge, in the existing research on arm exoskeleton assistance, no prior work has treated human-robot interaction torque as a state variable to establish a dynamic model and design a torque control algorithm based on it. There are two primary reasons for this gap. The first reason is the complexity of the physical characteristics of the human-robot interaction port, which makes it challenging to model the interaction torque. The second reason is that the interaction torque is influenced by the user's voluntary movements, meaning that the model inevitably contains information related to the user's voluntary movement trajectories, further complicating the modeling process.

    In this study, we were able to effectively address these two challenges, thanks to our extensive preparatory work and the limited application scenario. To tackle the challenge of the complex physical properties of the human-robot interaction port, we previously installed an SEA at the human-robot interaction port [1]. With this design, the SEA is essentially connected in series with the interaction port, such that its output torque closely approximates the human-robot interaction torque. Therefore, by modeling the SEA output torque, we can effectively model the interaction torque. To address the influence of voluntary human movement, we previously proposed an online prediction algorithm for point-to-point arm movement trajectories [2], which forms a crucial foundation for applying the proposed torque model within the RMPC framework presented in this paper.

    Most of the existing work on modeling human-robot interaction force to enhance force control accuracy has been focused on surgical robots. For example, Dominici *et al.* proposed a second-order hysteresis model for the interaction force and used model predictive control to achieve precise force control during cardiac surgery [3]. Similarly, Wijayarathne *et al.* developed a complex interaction force model with numerous physical parameters for a surgical robot [4]. Further examples of similar work can be found in the review article [5]. There are two main differences between these studies and our work. First, the interaction force models used for surgical robots are typically complex nonlinear models. While more complex and accurate modeling can improve force control precision, it also increases the complexity of

the algorithm and lowers the control frequency. Second, the surgical robot interacts with a static environment, and thus the force model does not need to account for voluntary human movements. From this perspective, designing the interaction force model for exoskeletons is relatively more complex.

In summary, the idea of improving torque control accuracy through a human-robot interaction torque model in this study draws inspiration from the research on surgical robots. Moreover, we have adapted the model to suit the specific requirements of exoskeleton assistance scenarios. In addition, to more clearly illustrate the similarities and differences between our research and the other research on human-robot interaction force modeling, we have revised the Introduction section of the paper, improving the logical flow of the relevant content to make it easier for readers to understand our work, as highlighted in red below.

[1] X. Xia, L. Han, H. Li, Y. Zhang, Z. Liu, L. Cheng, "A compliant elbow exoskeleton with an SEA at interaction port," in *Proceedings of the 2023 International Conference on Neural Information Processing*, Changsha, China, vol. 14450, pp. 146-157, 2023.
[2] X. Xia, L. Cheng, M. Ma, H. Zhang, L. Han, and H. Li, "Neurologically inspired transparent interaction for wearable exoskeletons," *IEEE/ASME Transactions on Mechatronics*, DOI: 10.1109/TMECH.2024.3492010, early access.
[3] M. Dominici and R. Cortesao, "Model predictive control architectures with force feedback for robotic-assisted beating heart surgery," in *Proceedings of IEEE International Conference on Robotics and Automation*, Hong Kong, China, 2014, pp. 2276-2282.
[4] L. Wijayarathne, Z.Y. Zhou, Y. Zhao, and F.H. Hammond, "Real-time deformable-contact-aware model predictive control for force-modulated manipulation," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3549-3566, 2023.
[5] A.S.S. Abadi, A. Ordys, K. Kukielka, and B. Pierscionek, "Review on challenges for robotic eye surgery; surgical systems, technologies, cost-effectiveness, and controllers," International Journal of Medical Robotics and Computer Assisted Surgery, vol. 19, no. 4, pp. 1-18, 2023.

In addition to exoskeletons, surgical robot control scenarios also treat the interaction force as the controlled variable. In such scenarios, precise control of the human-robot interaction force is crucial. Therefore, the physical characteristics of the human-robot interaction port are typically modeled. It is worth mentioning that the concept of the human-robot interaction port typically refers to the area where the human and the robot make contact [24]. For example, Dominici *et al.* proposed a second-order hysteresis model for the interaction force [25] and applied model predictive control to achieve precise force control during cardiac surgery. Similarly, Wijayarathne *et al.* developed an interaction force model with numerous physical parameters for a surgical robot [26]. The main drawbacks of these models are their high complexity and the need to identify numerous physical parameters, which can reduce the real-time control frequency of the algorithm.

However, even if the control frequency issue can be resolved, applying the interaction force modeling approaches used in surgical robots to exoskeletons still faces two key challenges. First, surgical robots interact with flexible human tissue through a rigid tip, resulting in a small contact area that makes nonlinear modeling of the human-robot interaction force feasible. In contrast, the human-exoskeleton interaction port covers a larger area and typically involves flexible materials such as straps, making direct nonlinear physical modeling of the interaction force much less feasible. Second, the interacted object in surgical robots is static. In exoskeleton applications, however, the user actively moves. In this case, accurately modeling the human-robot interaction force would require predicting the user's movements.

– ***Comment 2:*** *How would you conduct a comparative analysis of the ARMPC, RMPC, and IC control methods? Could you specify the evaluation metrics employed in this comparative study? It is recommended to provide a detailed elaboration.*

Thank you for your careful review. Overall, in the revised manuscript, we primarily adopted two evaluation metrics—torque tracking error and settling time—to compare the controllers. This choice is motivated by the objective of this study: to develop a control algorithm capable of accurately tracking the reference human-robot interaction torque, which serves as the foundation for designing and validating the exoskeleton's assistance strategy. Therefore, the torque tracking accuracy of the controller is one of our key concerns.

Specifically, we provide the numerical results and corresponding analysis of these metrics after each simulation and experiment. For instance, in the simulation comparing the trajectory tracking performance of different controllers, we present the following tracking error values along with analysis:

The mean absolute errors (MAE) in torque trajectory tracking for ARMPC, RMPC, Tube-based MPC, and IC are 0.043 Nm, 0.052 Nm, 0.661 Nm, and 0.203 Nm, respectively. It can be observed that the control algorithms based on the interaction torque model (ARMPC, RMPC, and Tube-based MPC) exhibit higher control accuracy, demonstrating that the proposed model effectively captures the characteristics of the human-robot interaction torque. Moreover, ARMPC achieves the highest control accuracy, indicating that the adaptive algorithm provides more reasonable feedback controller parameters under different error states than RMPC. This improvement is beneficial for the design and validation of exoskeleton assistance strategies.

In the simulation involving a sudden and large disturbance, we provide the following settling time values and analysis:

From the simulation results, it can be seen that all controllers can stably continue tracking the interaction torque trajectory. To quantitatively analyze the controllers' performance, we define the settling time as the duration required for the interaction torque to return to the allowable error band ($\pm 0.15$ Nm). The settling times for ARMPC, RMPC, Tube-based MPC,

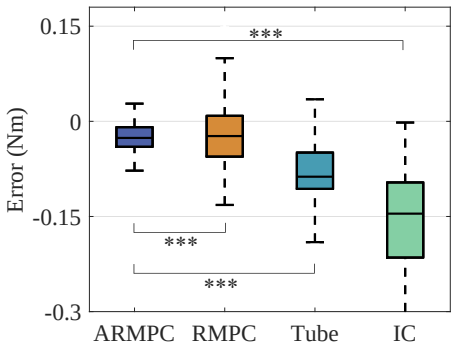



Fig. 10: Torque tracking errors comparison. The 'Tube' refers to the Tube-based MPC.

and IC are 0.09 s, 0.11 s, 0.13 s, and 0.18 s, respectively. ARMPC achieves the shortest settling time, which can be attributed to the inclusion of an adaptive error feedback controller.

In the experiment comparing the trajectory tracking performance of different controllers, the specific tracking MAE values are provided in Table I of the revised manuscript, and the relative analysis is shown below. This section includes a discussion on the relationship between control accuracy and robustness, enhancing the comprehensiveness and completeness of the analysis.

From Fig. 9, it can be seen that the experimental results align closely with the simulation in Section V, with ARMPC showing the best performance. This further confirms that the introduction of the interaction torque model and the adaptive algorithm can effectively enhance the precision of the interaction torque controller. Video 1 in the supplementary materials demonstrates the same results. It is worth noting that there is often a trade-off between the controller precision and the robustness. Theoretically, Tube-based MPC improves robustness at the expense of control accuracy. Similarly, the adaptive algorithm proposed in this paper may also reduce the controller's robustness to some extent. Since it is difficult to precisely control the magnitude of external disturbance torques in physical experiments, the controller's response to large disturbances is discussed through simulations in Section V to demonstrate that the controller has sufficient robustness. In the real-world experiments, the exoskeleton assistance tasks were limited to load-lifting activities, and all participants were healthy individuals. As a result, occurrences of sudden and significant disturbances were extremely rare. Moreover, to ensure user safety, the human-robot interaction force was monitored in real time during the experiments. If the interaction force exceeded a predefined threshold, the exoskeleton would be shut down to prevent potential harm.

In addition, regarding the metric of interaction torque tracking error, statistical analysis has been added in the revised manuscript, as shown below.

Considering that the results in Fig. 9 are approximate, for each controller, the control

errors from ten subjects were combined and visualized using box plots, as shown in Fig. 10. *** in Fig. 10 indicates a statistically significant difference between the two groups (t-test, $p < 0.001$). It can be observed that the interaction torque control error of ARMPC is significantly lower than that of the other controllers.

– ***Comment 3:*** *Please provide a theoretical explanation of the comparative results among the three control methods.*

Thank you for your careful review. In the revised manuscript, we have added more theoretical analyses to explain why the proposed ARMPC can improve control accuracy. For example, in the analysis of the simulation results comparing the torque tracking performance of different controllers, we supplemented the following theoretical explanation:

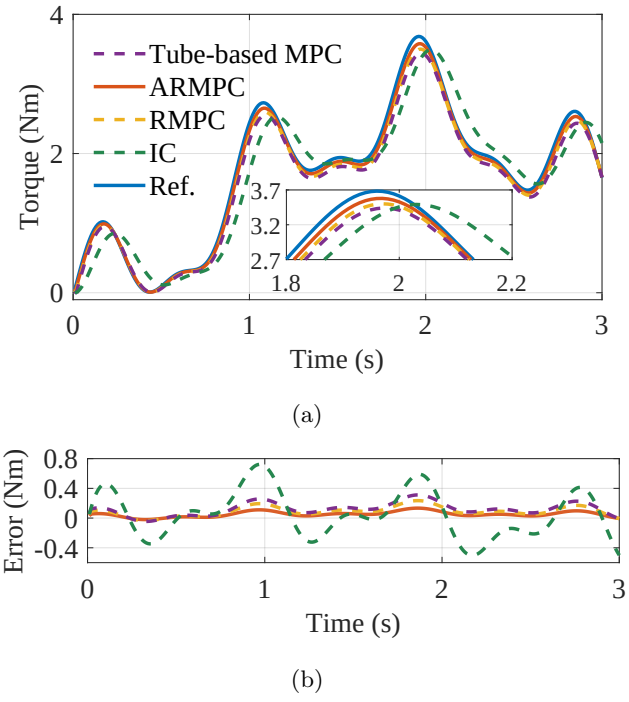
It can be observed that the control algorithms based on the interaction torque model (ARMPC, RMPC, and Tube-based MPC) exhibit higher control accuracy, demonstrating that the proposed model effectively captures the characteristics of the human-robot interaction torque. Moreover, ARMPC achieves the highest control accuracy, indicating that the adaptive algorithm provides more reasonable feedback controller parameters under different error states than RMPC. This improvement is beneficial for the design and validation of exoskeleton assistance strategies. It is also worth noting that the Tube-based MPC is a more conservative algorithm that sacrifices some accuracy to improve controller robustness.

In addition, the relationship between control accuracy and robustness was discussed as a supplementary theoretical explanation of the experimental results.

It is worth noting that there is often a trade-off between the controller precision and the robustness. Theoretically, Tube-based MPC improves robustness at the expense of control accuracy. Similarly, the adaptive algorithm proposed in this paper may also reduce the controller's robustness to some extent. Since it is difficult to precisely control the magnitude of external disturbance torques in physical experiments, the controller's response to large disturbances is discussed through simulations in Section V to demonstrate that the controller has sufficient robustness. In the real-world experiments, the exoskeleton assistance tasks were limited to load-lifting activities, and all participants were healthy individuals. As a result, occurrences of sudden and significant disturbances were extremely rare. Moreover, to ensure user safety, the human-robot interaction force was monitored in real time during the experiments. If the interaction force exceeded a predefined threshold, the exoskeleton would be shut down to prevent potential harm.

– ***Comment 4:*** *Please add a relatively complex torque reference experiment instead of a sinusoidal torque reference.*

Thank you for your careful review. A relatively complex torque reference has been designed




(a)




(b)

Fig. 7: The performance of the proposed torque controllers in simulation.

and validated in the revised manuscript. The expression of the complex torque reference is

$$r_1 = 1.5\sin(1.39t - 0.94) + 0.75\sin(6.95t + 0.63) + 0.3\sin(13.9t - 0.31) + 1.14.$$

The validation is conducted through simulation. We assumed that the user's arm remains stationary in a horizontal position. And the supplementary validation and analysis in the revised manuscript are as follows.

In addition, we further validated the controller's ability to track complex torque trajectories. We assumed that the user's arm remains stationary in a horizontal position and designed the following complex torque reference trajectory:

$$r_1 = 1.5\sin(1.39t - 0.94) + 0.75\sin(6.95t + 0.63) + 0.3\sin(13.9t - 0.31) + 1.14. \quad (62)$$

The tracking performance for the complex torque trajectory is shown in Fig. 7. The mean absolute tracking errors for ARMPC, RMPC, Tube-based MPC, and IC are 0.053 Nm, 0.084 Nm, 0.125 Nm, and 0.255 Nm, respectively. These results demonstrate that all the above control algorithms can effectively track complex trajectories, with the ranking of tracking accuracy consistent with the earlier simulation results.

– ***Comment 5:*** *If there is a strong disturbance in the experiment, how will the results of the three control methods change?*

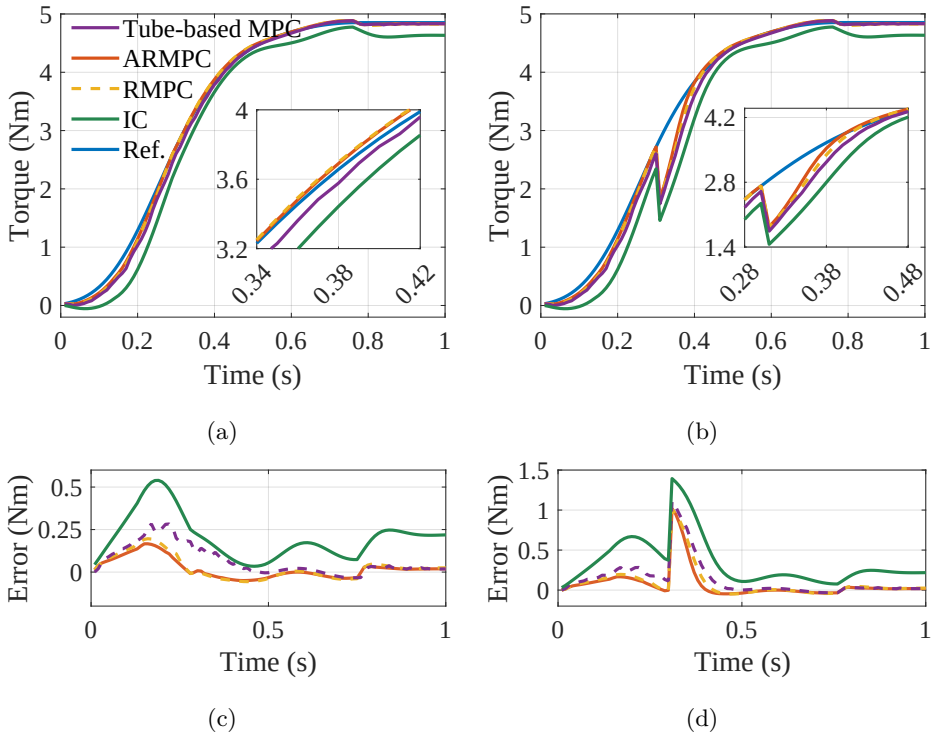



Fig. 6: The performance of the proposed torque controllers in simulation. (a) The torque tracking performance of the controllers. (b) The torque tracking performance in the presence of the sudden disturbance. (c) The torque tracking error of the controllers. (d) The torque tracking error in the presence of the sudden disturbance.

Thank you for your careful review. Considering that it is difficult to precisely control the magnitude of external disturbance torques in physical experiments, the controller's response to sudden, large disturbances is validated through simulation. Specifically, a sudden disturbance torque of 1 Nm was introduced in the simulation to evaluate the performance of each controller. By comparing the tracking errors and the settling time of each controller, we assessed their performance under the sudden perturbation. The simulation results show that all controllers were able to continue tracking the reference torque in a stable manner. Among them, the proposed ARMPC achieved the shortest settling time. The simulation results and corresponding analysis in the revised manuscript are shown below.

It is also worth noting that the Tube-based MPC is a more conservative algorithm that sacrifices some accuracy to improve controller robustness. Therefore, to verify whether the proposed controller can withstand larger disturbances, we introduced a 1 Nm disturbance at 0.3 seconds in the simulation to mimic user-induced errors due to sudden tremors. The simulation results after introducing the 1 Nm disturbance are shown in Fig. 6(b) and Fig. 6(d).

From the simulation results, it can be seen that all controllers can stably continue tracking the interaction torque trajectory. To quantitatively analyze the controllers' performance, we define the settling time as the duration required for the interaction torque to return to the allowable error band (±0.15 Nm). The settling times for ARMPC, RMPC, Tube-based MPC, and IC are 0.09 s, 0.11 s, 0.13 s, and 0.18 s, respectively. ARMPC achieves the shortest settling time, which can be attributed to the inclusion of an adaptive error feedback controller.

– ***Comment 6:*** *Which references do the two comparative control methods come from? Is that the latest research? Please point out. The formulas of these two control methods do not appear in the paper.*

Thank you for your careful review. In this study, RMPC and impedance control were employed as baseline algorithms for comparison. The RMPC approach is derived from several recent state-of-the-art studies [6], [7]. For example, Xu *et al.*, in their 2025 paper published in *IEEE Transactions on Cybernetics*, developed an RMPC method based on a linearized system dynamics model [6]. This demonstrates that RMPC is a commonly used control algorithm in advanced exoskeleton research.

Impedance control, on the other hand, is one of the most widely used force control strategies in exoskeleton robots. It enables compliant human-robot interaction without requiring explicit modeling of the interaction force. These advantages have led to its widespread adoption and continued application in recent advanced studies [8], [9].

Given the advanced nature and broad application of these two methods in exoskeleton control, they were selected as comparative baselines in our work. In the revised manuscript, we have added detailed explanations of the advanced nature of both algorithms, along with their specific formulations. Furthermore, to enhance the completeness of the comparative analysis, we have also introduced the Tube-based MPC as another advanced control strategy. The revised content in the manuscript is presented as follows.

[6] L.R. Xu, Z.J. Li, G.X. Li, and L.J. Jin, "Robust model predictive control of a gait rehabilitation exoskeleton with whole body motion planning and neuro-dynamics optimization," *IEEE Transactions on Cybernetics*, DOI: 10.1109/TCYB.2025.3545064, early access.

[7] Z.Y. Sun, A. Iyer, K. Lambeth, C. Cleveland, and N. Sharma, "Knee extension tracking and fatigue regulation results using a robust MPC approach in a hybrid exoskeleton," *Control Engineering Practice*, vol. 141, no. 105717, 2023.

[8] L. Liu, M. Illian, S. Leonhardt, and B.J.E. Misgeld, "Iterative learning control for cascaded impedance-controlled compliant exoskeleton with adaptive reaction to spasticity," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, article number 4008211, 2023.

[9] X.F. Xiong, C.D. Do, and P. Manoonpong, "Learning-Based Multifunctional Elbow Exoskeleton Control," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 9, pp. 9216-9224, 2022.

The analysis of the algorithm's advanced nature is presented as follows:

In recent exoskeleton research, robust model predictive controllers (RMPC) have been widely used [35]-[37]. For example, Xu *et al.* designed an RMPC based on a linearized system dynamic model [35]. Bao *et al.* developed a control system for a knee exoskeleton using a Tube-based MPC approach [37]. Compared to these studies, our work differs in two main ways. First, we propose a dynamic model for human-robot interaction torque and incorporate it into the model predictive control framework. Second, we integrate an adaptive strategy into the RMPC to enhance the controller's accuracy. Given the advanced nature of these algorithms, we included a comparative simulation with both RMPC and Tube-based MPC to highlight the advantages of our proposed ARMPC. The RMPC used for comparison is similar to the proposed ARMPC, except that its $K_P$ and $K_I$ in (40) are set to appropriate fixed values. The Tube-based MPC used for comparison is a more robust MPC algorithm that leverages error tubes and tightened constraints, with specific design methods detailed in [37].

In addition, impedance control (IC) is one of the most widely used force control algorithm in exoskeleton robots. It can achieve compliant human-robot interaction without the need to model the interaction forces, making it widely adopted in exoskeleton assistance [38], [39]. Consequently, this study also uses impedance control as a comparison algorithm.

The supplementary specific formulations are as follows:

The RMPC used for comparison is similar to the proposed ARMPC, except that its $K_P$ and $K_I$ in (40) are set to appropriate fixed values. The Tube-based MPC used for comparison is a more robust MPC algorithm that leverages error tubes and tightened constraints, with specific design methods detailed in [37].

Consequently, this study also uses impedance control as a comparison algorithm. Here, the desired trajectory $q_d$ in the impedance control is set to $q_h$. The desired second-order impedance model is designed as follows:

$$M_d\left(\ddot{q}-\ddot{q}_h\right)+D_d\left(\dot{q}-\dot{q}_h\right)+K_d\left(q-q_h\right)=r_1-\tau_i, \tag{58}$$

where $M_d$, $D_d$, and $K_d$ represent the desired mass, damping, and stiffness of the impedance model, respectively. $r_1$ is the desired interaction torque. Then (58) can be simplified to the following form:

$$\ddot{q}=\ddot{q}_h+M_d^{-1}\left(r_1-\tau_i-D_d\left(\dot{q}-\dot{q}_h\right)-K_d\left(q-q_h\right)\right). \tag{59}$$

Combining (59) and (4), the following impedance controller is designed:

$$\tau_m=M_r\left(\ddot{q}_d+M_d^{-1}\left(r_1-\tau_i-D_d\left(\dot{q}-\dot{q}_h\right)-K_d\left(q-q_h\right)\right)\right)+C_r\dot{q}+G_r+K_f\left(r_1-\tau_i\right), \tag{60}$$

where $K_f$ is the torque error feedback gain.

The supplementary Tube-based MPC for comparison is as follows:

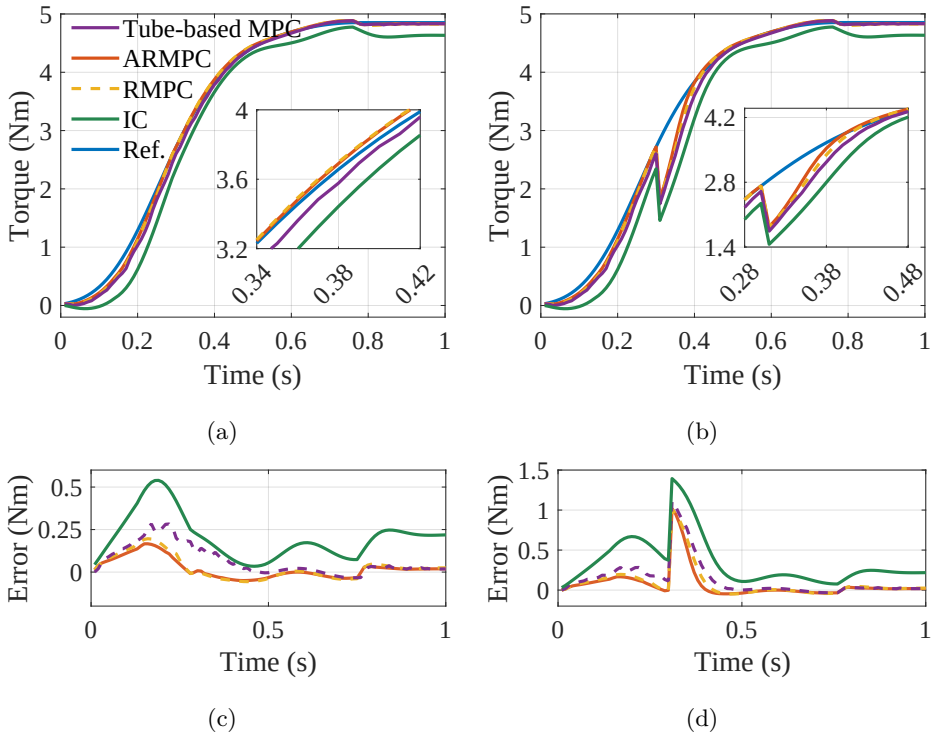



Fig. 6: The performance of the proposed torque controllers in simulation. (a) The torque tracking performance of the controllers. (b) The torque tracking performance in the presence of the sudden disturbance. (c) The torque tracking error of the controllers. (d) The torque tracking error in the presence of the sudden disturbance.

The simulation results are shown in Fig. 6(a) and Fig. 6(c). The mean absolute errors in torque trajectory tracking for ARMPC, RMPC, Tube-based MPC, and IC are 0.043 Nm, 0.052 Nm, 0.661 Nm, and 0.203 Nm, respectively. It can be observed that the control algorithms based on the interaction torque model (ARMPC, RMPC, and Tube-based MPC) exhibit higher control accuracy, demonstrating that the proposed model effectively captures the characteristics of the human-robot interaction torque. Moreover, ARMPC achieves the highest control accuracy, indicating that the adaptive algorithm provides more reasonable feedback controller parameters under different error states than RMPC. This improvement is beneficial for the design and validation of exoskeleton assistance strategies. It is also worth noting that the Tube-based MPC is a more conservative algorithm that sacrifices some accuracy to improve controller robustness.

- **Responses to Reviewer 4**
  - ***General Comments:*** *In this paper, the authors proposed a transparent lifting assistance strategy for arm exoskeletons using an adaptive torque controller. The reviewer has following comments.*

    **Thank you for your valuable comments and suggestions.** In the revised manuscript, all the comments have been carefully answered and addressed.

  - ***Comment 1:*** *What is the relationship between $d_1$ and $d_2$?*

    Thank you for your careful review. According to (3) in the manuscript, the relationship between $q$ and $\tau_i$ can be derived

    $$q = q_h - \frac{\tau_i}{K_s l^2} + \frac{d_1}{K_s l^2}.$$

    Substitute the relationship into (4) in the manuscript, the relationship between $d_1$ and $d_2$ can be derived as

    $$d_2 = \frac{M_r \ddot{d}_1}{K_s l^2} + \frac{B_r \dot{d}_1}{K_s l^2}.$$

    The relationship between $d_1$ and $d_2$ has been added in the revised manuscript.

  - ***Comment 2:*** *In (10), there is no known term $\frac{B_r H}{K_s l^2}$, and it cannot be canceled out by the term $\frac{B_r H}{K_s}$*

    Thank you for your careful review. There are some writing errors in this section, which have been corrected in the revised manuscript, as shown below.

    $$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{K_s l^2}{M_r} & -\frac{B_r}{M_r} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{K_s l^2}{M_r} \end{bmatrix} u + \begin{bmatrix} 0 \\ \frac{K_s l^2 H}{M_r} + \frac{K_s l^2 d_3}{M_r} \end{bmatrix}. \tag{11}$$

    Let $u = u' + H$, the known term $\frac{K_s l^2 H}{M_r}$ can be canceled out.

  - ***Comment 3:*** *In (23), the cost function $J$ is a function of $z$, then why in (30), $e$ instead of $z$ is involved?*

    Thank you for your careful review. **e** in (30) should be **z**, and these errors have been corrected in the revised manuscript, as shown below.

    According to the optimality of $V^*(k)$, the following inequality can be obtained.

    $$V^*(k+1) - V^*(k) \leq J(k+1) - V^*(k). \tag{31}$$

    Define the optimal control input sequence in $V^*(k)$ as $\left[v_1^*(0|k), \cdots, v_1^*(N-1|k)\right]$, and the control input sequence in $J(k+1)$ as $\left[v_1^*(1|k), \cdots, v_1^*(N-1|k), v_1(N|k)\right]$, where $v_1(N|k) =$

$K\mathbf{z}(N|k)$. Then (31) can be further simplified

$$\begin{aligned} V^*(k+1) - V^*(k) &\leq J(k+1) - V^*(k) \\ &= \|v_1(N|k)\|_R^2 + \|\mathbf{z}(N|k)\|_Q^2 + \|\mathbf{z}(N+1|k)\|_P^2 \\ &- \|\mathbf{z}(N|k)\|_P^2 - \|\mathbf{z}(1|k)\|_Q^2 - \left\|v_1^*(0|k)\right\|_R^2. \end{aligned} \tag{32}$$

– ***Comment 4:*** *In Section IV, it seems that in order to train the ANFIS NN, the authors tuned a set of $K_P$ and $K_I$ parameters using expert knowledge, and then the ANFIS NN is trained to approximate these parameters. Then why not using these $K_P$ and $K_I$ parameters directly in the the controller?*

Thank you for your careful review. In our study, the $K_P$ and $K_I$ parameter sets were initially determined based on expert knowledge, and the ANFIS NN was trained to approximate these values. While directly using the expert-tuned parameters is indeed a reasonable approach, we chose to employ the ANFIS NN for two main reasons. First, the expert-tuned parameters are sparse and typically designed for specific conditions. By using an ANFIS NN, the controller parameters can be more flexibly generalized across a wider range of applications. Second, although we employed B-spline equations to augment the sparse expert knowledge, we did not directly use B-spline equations as the model. This is mainly because B-spline expressions are relatively complex and less intuitive. In contrast, the ANFIS NN offers simpler expressions and stronger interpretability. Additionally, controllers based on ANFIS NN have better scalability, for example, allowing online parameter updates using reinforcement learning, which is also an important motivation for adopting the ANFIS NN approach.

To ensure that readers can fully understand the purpose of our training data design and to avoid any potential misinterpretation, we have rewritten the description of the training data collection and data augmentation process in the revised manuscript, as highlighted in red font below.

To generate the training data, the values of $K_P$ and $K_I$ are recorded for different error conditions based on the expert knowledge. The aforementioned different error conditions include what we consider the most extreme cases—*i.e.*, the cases with the maximum and minimum errors. Furthermore, sparse sampling is achieved through non-uniform sampling, where more sampling points are selected when the error is smaller. Based on these sampling points and expert knowledge, sparse training data points are obtained. The sparse training data points are then augmented with smooth transition functions.

To guarantee the error boundedness of the controller, parameters $K_P$ and $K_I$ in the training set are required to strictly satisfy the following conditions:

$$|\mathrm{eig}(A_c)| < 1, \tag{48}$$

$$A_c = \begin{bmatrix} A - \bar{B}K_P & -\bar{B}K_I \\ A - \bar{B}K_P & -\bar{B}K_I + I \end{bmatrix}, \tag{49}$$

where $I$ is the identity matrix. Condition (48) essentially provides the necessary and sufficient condition for the stability of the closed-loop discrete-time linear system with a proportional-integral (PI) controller. To ensure that all points in the training set satisfy (48), the $K_P$ and $K_I$ designed based on expert knowledge are conservative. In addition, we extracted the extreme points from the augmented training set and confirmed that these points also satisfy (48), thereby ensuring that the entire training set meets this condition.

– ***Comment 5:*** *In (16), the reference torque trajectory can only be generated based on the known weight of the lifted object. However, in practice, it is usually not possible to know the weight of the object in advance. Thus the presented method seems impractical.*

Thank you for your careful review. To achieve the torque assistance with a exactly given $\gamma$, the weight of the lifted object should be known previously.

However, in practical applications, such as cargo handling at airport, the approximate weight range of the load is usually known, while its exact mass is often unavailable. In such cases, the parameter $\gamma G$ can be set to a fixed value. Ideally, this configuration effectively reduces the perceived weight of the load by $\gamma G$ kg, rather than by $\gamma \times 100\%$. As a result, the assisted user still perceives lifting a lighter object and can maintain their natural exertion patterns. Therefore, the proposed algorithm remains practically useful even when the precise mass of the load is unknown.

For the convenience of readers' understanding, the revised manuscript has added a discussion on this situation, as shown below.

In practical applications, if the weight of the object is unavailable, the parameter $\gamma G$ can be set to a fixed and appropriate value. Ideally, this configuration reduces the perceived weight of the load by $\gamma G$ kg, rather than by $\gamma \times 100\%$. But it can also maintain the natural exertion patterns of the user.

– ***Comment 6:*** *What is the output of the prediction algorithm in [25]? If $\hat{q}_h$ is the output, then how to obtain $\dot{\hat{q}}_h$ and $\ddot{\hat{q}}_h$?*

Thank you for your careful review. In [25] ([28] in the revised manuscript), the trajectories are modeled by the Probabilistic Movement Primitives (ProMP) algorithm, and the expression of the ProMP model is

$$\begin{bmatrix} \hat{q}_h & \dot{\hat{q}}_h \end{bmatrix}^T = \begin{bmatrix} \phi & \dot{\phi} \end{bmatrix}^T \theta + \varepsilon,$$

where $\phi \in \mathbb{R}^{n\times 1}$ is the time-dependent basis function, $n$ is the number of the basis functions, $\theta \in \mathbb{R}^{n\times 1}$ represents the weights of the basis functions, and $\varepsilon$ is a zero-mean Gaussian noise term. As a result, the $\hat{q}_h$ and $\dot{\hat{q}}_h$ are both outputs of the ProMP model, and they can be obtained directly. Then $\ddot{\hat{q}}_h$ can be calculated through the difference processing of $\dot{\hat{q}}_h$. For the

convenience of readers' understanding, a further explanation has been added in the revised manuscript, as shown below.

In our previous work, a neurology-inspired algorithm for arm movement prediction is proposed [28]. The algorithm can predict the trajectories of $q_h$ and $\dot{q}_h$, and the value of $\ddot{q}_h$ can be calculated through the difference processing of $\dot{q}_h$.

– ***Comment 7:*** *Is it really necessary to build the dynamic model given in (10)? Why tracking the trajectory given in (16) based on the dynamic model is better than just providing the assistance lifting torque based on the angle $\theta$ directly? For example, you can direct output a torque $\tau_m = \gamma\,(GL + G_r)\,sin\,(\theta)$ using the motor.*

Thank you for your careful review. The dynamic model is necessary for improving torque tracking accuracy, and there are two main reasons for this.

The first reason is that the dynamic model proposed in this paper incorporates predictive information about human arm movement trajectories, which is crucial for enhancing torque tracking accuracy. If the exoskeleton's interaction target were a rigid, static object, directly adjusting the motor's output torque might suffice to achieve fast and relatively precise interaction torque control. However, in this study, the exoskeleton interacts with an actively moving human arm. Even if the exoskeleton's output torque is set to zero, active arm movements can still result in unintended interaction torques. Moreover, directly commanding the exoskeleton to output the desired torque would still lead to relatively low tracking accuracy. In this context, accurately predicting and following the user's arm movements is equivalent to making the exoskeleton and the arm quasi-static relative to each other, providing a stable basis for applying interaction torques and significantly improving the torque control accuracy.

The second reason is that the dynamic model incorporates physical characteristics of the human-exoskeleton interaction port. In surgical robotics, modeling the interaction port is an important approach for improving the torque control accuracy [1], [2]. In this study, the springs in the SEA are essentially in series with the user's arm. Therefore, modeling the physical properties of the SEA effectively represents the characteristics of the human-exoskeleton interaction port. Designing a torque controller based on this model can help improve the control accuracy.

[1] M. Dominici and R. Cortesao, "Model predictive control architectures with force feedback for robotic-assisted beating heart surgery," in *Proceedings of IEEE International Conference on Robotics and Automation*, Hong Kong, China, 2014, pp. 2276-2282.
[2] L. Wijayarathne, Z.Y. Zhou, Y. Zhao, and F.H. Hammond, "Real-time deformable-contact-aware model predictive control for force-modulated manipulation," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3549-3566, 2023.

– ***Comment 8:*** *Why the sEMG signal with the assistance of the exoskeleton should be the close*

*to the sEMG signal without the assistance of the exoskeleton? When the user is providing different effort, the sEMG signal should also be different, isn't it?*

Thank you for your careful review. If the assistance tasks are constantly varying, the comparison between sEMG signals is meaningless in the context of this paper. For example, the sEMG signals recorded during lifting tasks exhibit marked differences from those acquired during push-up tasks.

However, to ensure meaningful comparisons between sEMG signals, this paper implements standardized assistance tasks. Specifically, both the movement range and cycle duration are strictly maintained during dumbbell lifting experiments. As a result, the comparisons between sEMG signals before and after wearing exoskeletons can reveal purposeful biomechanical information.

sEMG signals directly correlate with muscle activation levels and essentially reflect muscular force output [3]. In other words, the amplitude cures of sEMG signals effectively represent arm exertion patterns. It should be noted that unreasonable assistance strategies may induce parasitic forces [4]. To compensate for these parasitic forces, the shape of the user's exertion curve will change. This change can cause discomfort for users, reduce their adaptability to the exoskeleton, and even affect their standing balance. Therefore, in the same task, comparing the similarity of sEMG signals can demonstrate the effectiveness of the assistance strategy to some extent.

[3] D.G. Lloyd and T.F. Besier, "An EMG-driven musculoskeletal model to estimate muscle forces and knee joint moments in vivo," *Journal of Biomechanics*, vol. 36, pp. 765-776, 2003.
[4] I. Pacifico, A. Scano, E. Guanziroli, M. Moise, L. Morelli, A. Chiavenna, D. Romo, S. Spada, G. Colombina, F. Molteni, F. Giovacchini, N. Vitiello, and S. Crea, "An experimental evaluation of the Proto-MATE: A novel ergonomic upper-limb exoskeleton to reduce workers' physical strain," *IEEE Robotics and Automation Magazine*, vol. 27, no. 1, pp. 54-65, 2020.

– ***Comment 9:*** *There is an extra "," in (27).*

Thank you for your careful review. The extra "," in the equation has been removed in the revised manuscript.

- **Responses to Reviewer 5**
  - ***General Comments:*** *This manuscript presents a highly meaningful study, demonstrating clear logical thinking and substantial work content. However, to further enhance its quality, the following recommendations deserve your careful attention:*

    **Thank you for your valuable comments and suggestions.** In the revised manuscript, all the comments have been carefully answered and addressed.

  - ***Comment 1:*** *To better facilitate reader comprehension, please provide a detailed derivation of Equation (30).*

    Thank you for your valuable comment. Prior to presenting the detailed derivation of Equation (30) in the original manuscript, we have rectified a typographical error in its original expression by replacing the erroneous variable '**e**' with the correct notation '**z**'. The first line of the Equation (30) in the original manuscript is

$$V^*(k+1) - V^*(k) \leq J(k+1) - V^*(k).$$

    Define the optimal control input sequence in $V^*(k)$ as

$$\left[v_1^*(0|k), v_1^*(1|k), \cdots, v_1^*(N-1|k)\right],$$

    and the control input sequence in $J(k+1)$ as

$$\left[v_1^*(1|k), v_1^*(2|k), \cdots, v_1^*(N-1|k), v_1(N|k)\right],$$

    where the superscript * represents the optimality of the control input. The mathematical expressions of $V^*(k)$ and $J(k+1)$ are subsequently derived as follows.

$$V^*(k) = \sum_{n=1}^{N-1} \|\mathbf{z}(n|k)\|_Q^2 + \sum_{m=0}^{N-1} \left\|v_1^*(m|k)\right\|_R^2 + \|\mathbf{z}(N|k)\|_P^2,$$

$$J(k+1) = \sum_{n=2}^{N} \|\mathbf{z}(n|k)\|_Q^2 + \sum_{m=1}^{N-1} \left\|v_1^*(m|k)\right\|_R^2 + \|v_1(N|k)\|_R^2 + \|\mathbf{z}(N+1|k)\|_P^2.$$

    Then we can derived that

$$\begin{aligned} J(k+1) - V^*(k) = {} & \|v_1(N|k)\|_R^2 + \|\mathbf{z}(N|k)\|_Q^2 + \|\mathbf{z}(N+1|k)\|_P^2 - \|\mathbf{z}(N|k)\|_P^2 \\ & - \|\mathbf{z}(1|k)\|_Q^2 - \left\|v_1^*(0|k)\right\|_R^2. \end{aligned}$$

    To improve the clarity of the presentation, Equation (30) in the original manuscript has been reconstructed in the revised manuscript. The refined formulation is detailed as follows:

    According to the optimality of $V^*(k)$, the following inequality can be obtained.

$$V^*(k+1) - V^*(k) \leq J(k+1) - V^*(k). \tag{31}$$

Define the optimal control input sequence in $V^*(k)$ as $\left[v_1^*(0|k), \cdots, v_1^*(N-1|k)\right]$, and the control input sequence in $J(k+1)$ as $\left[v_1^*(1|k), \cdots, v_1^*(N-1|k), v_1(N|k)\right]$, where $v_1(N|k) = K\mathbf{z}(N|k)$. Then (31) can be further simplified

$$\begin{aligned} & V^*(k+1) - V^*(k) \leq J(k+1) - V^*(k) \\ & = \|v_1(N|k)\|_R^2 + \|\mathbf{z}(N|k)\|_Q^2 + \|\mathbf{z}(N+1|k)\|_P^2 \\ & - \|\mathbf{z}(N|k)\|_P^2 - \|\mathbf{z}(1|k)\|_Q^2 - \left\|v_1^*(0|k)\right\|_R^2. \end{aligned} \tag{32}$$

– ***Comment 2:*** *Some of the figures in the paper are relatively small, which may hinder interpretation. It is suggested to enlarge figures such as Fig. 5, 8, and 10 for improved clarity.*

Thank you for your careful review. For the convenience of reading, the figures have been enlarged in the revised manuscript.

**Again, we sincerely appreciate the Editors and Reviewers for your time and effort in reviewing this paper. Thank you very much! If you have further questions, please feel free to let us know.**